Ensemble BERT with Data Augmentation and Linguistic Knowledge on SQuAD 2.0

Wen Zhou, Hang Jiang, Xianzhe Zhang
{zhouwen, hjian42, xianzhez} @ stanford.edu

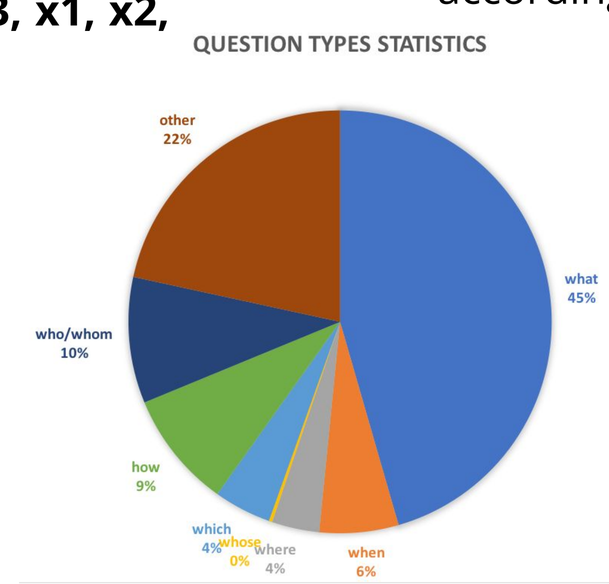
Problem

The official Stanford Question Answering Dataset (SQuAD) collects 100k question/answer pairs and set up a leaderboard to attract researchers to address this challenge. Based on SQuAD 1.1, SQuAD 2.0 added more questions that could not be answered according to the provided paragraphs. BERT, as one of the state-of-art language model, performs very well on SQuAD 2.0. But we think there is more work we can do that could improve the performance of BERT on question answering task especially on SQuAD 2.0. At the same time, the limitation of the volume of the data could also be a barrier to train a more effective model. We adopted data augmentation to enlarge our training data based on SQuAD.

SQuAD2.0
The Stanford Question Answering Dataset

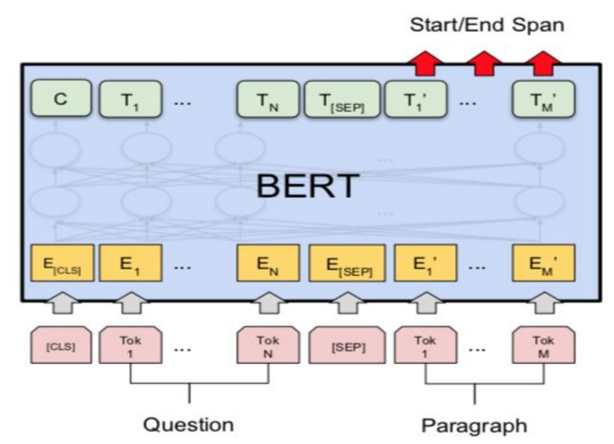
Data

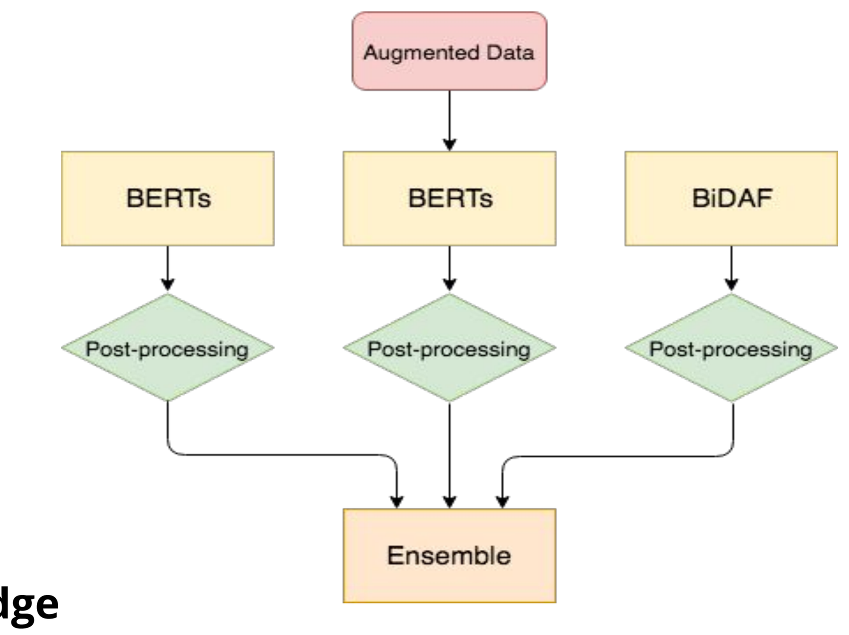
- **Stanford Question Answering Dataset (SQuAD) V2.0**: the input to the model is a question with a context paragraph, and the output should be the the span of text in the paragraph that can answer the question. There are about half of the questions in the dataset that cannot be answered given the provided paragraph. We used 129,941 examples as the training set, 6078 examples as the dev set, and 5915 examples as the test set.
- **Data Augmentation:** we perform synonym and random word replacement with NLTK and WordNet on the contexts of the SQuAD dataset. Questions are left unchanged. We explored different strategies of synonym replacement (**sampling rate, +random words, +stop words**) and injected different amount of augmented data (**x0.33, x1, x2, x3**) on top of the original data in our experiments.
- For each word in a context paragraph
  - 20% of the time:
    call replace_synonym
    - if exists synonyms:
      replace with a random synonym
    - otherwise:
      replace with a random word
  - 80% of the time: remain unchanged



Approach

1. **Baseline: BERT**
   a. BERT is a multi-layer bidirectional Transformer encoder with two novel unsupervised pre-training tasks (MLM and NSP) and constructed input representation. In terms of input, BERT constructs input representation by summing the WordPiece token embeddings, segment embeddings, and positional embeddings.
   b. Adaptation to SQuAD:





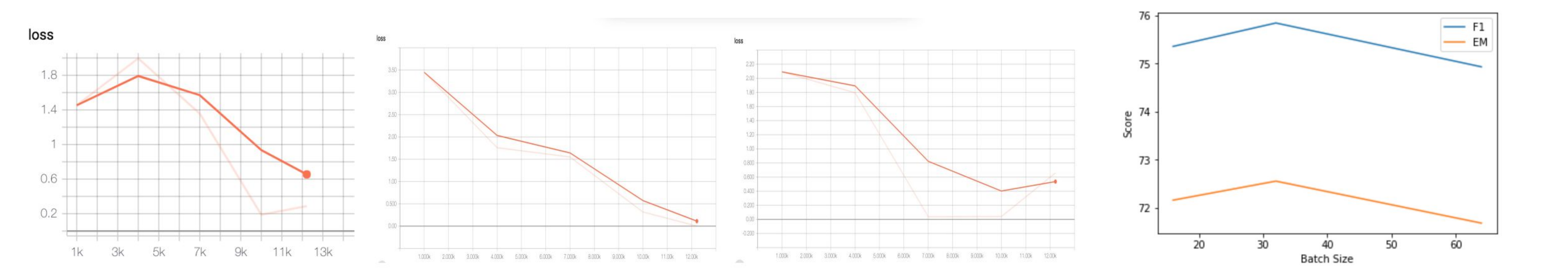
2. **Post-processing with Linguistic Knowledge**
   During prediction time, the probability for a text span Text(i, j) being the answer is:
   $$P(i,j) = \text{softmax}(\text{start\_logit}(i) + \text{end\_logit}(j))$$
   a. for "When" questions: if Text(i) is one of ['before', 'after', 'about', 'around', 'from', 'during'], add 0.2 to P(i, j);
   b. for "Where" questions: if Text(i) is one of ['in', 'at', 'on', 'behind', 'from', 'through', 'between', 'throughout'], add 0.2 to P(i, j);
   c. for "Whose" questions: if " 's " is contained in Text(i,j), add 0.2 to P(i, j);
   d. for "Which" questions: if Text(i) is "the", add 0.2 to P(i, j).
3. **Ensemble**
   For each question, we output the n-best predictions made by multiple models along with the probability, then for each proposed prediction, we sum up the probability from each model together, and finally output the prediction with the highest probability as the answer to that question. A weighting scheme is also used according to the performance of individual models.

Results

model	dev_F1	dev_EM
best finetuned BERT_base	75.841	72.557
best finetuned BERT_large	80.560	77.345
Ens1: 6 base BERTs	77.352	74.515
Ens2: 6 base BERTs + 0.2 BiDAF	77.362	74.580
Ens3: 7 base BERTs + 1 large BERT + 0.2 BiDAF	78.479	75.699
Ens4: 7 base BERTs + 1 large BERT + Weighting + 0.2 BiDAF	80.771	78.019
Ens5: 7 base BERTs + 3 large BERTs + Weighting + 0.2 BiDAF	81.666	79.154
Ens6: Ens5 + postprocessing with linguistic knowledge	81.764	79.434
Ens7: Ens6 + data augmentation	82.427	79.911
	test_F1	**test_EM**
Ens5	81.518	78.664
Ens7: Ranked top 1 on both dev and test leaderboards (until Mar.20 2AM)!!!	**82.540**	**79.865**

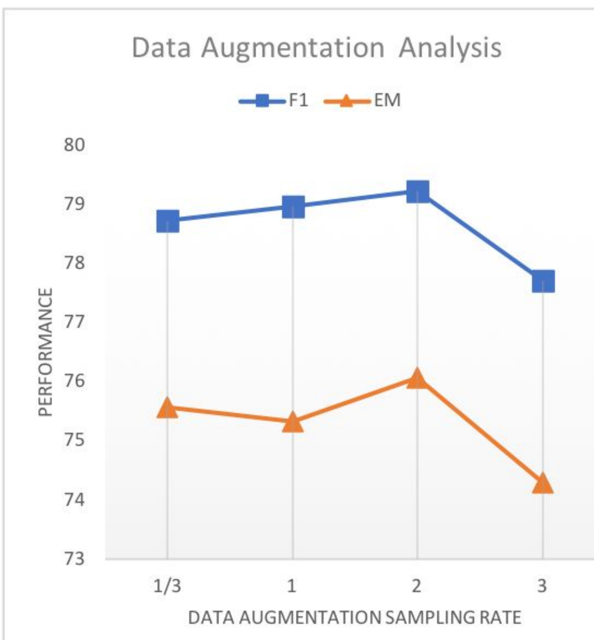
Analysis

- Hyper-parameter Search: we found 3E-05 is the best learning rate for this setting. For base BERT model, we found that batch size 32, epoch 3 worked the best; for large BERT model, we found that batch size 24, epoch 2 worked the best. This is because large model is more inclined to overfitting.



1) LR = 2E-05 — Too small: not converge
2) LR = 3E-05 — Good LR
3) LR = 5E-05 — Too large: overshoot optimum
Score vs. Batch Size (Base, seq_len=384, lr=3E-5, 3 epochs)

- Data Augmentation: injecting more augmented data generally performs better. The best observed sampling ratio is 1:2 (original:augmented). Removing random word replacement and checking stop words can improve the quality of the augmented data.
- Post-processing: adding linguistic heuristics can significantly boost EM, while slightly boost F1, meaning that the rules are good at precision (Ens5 vs Ens6)
- Ensemble: the best ensemble model can out-perform the best single model (#16) by 1.2 F1 and 2.1 EM. And a proper weighting scheme is very useful (Ens3 vs ENs4).



Conclusions

In our project, In our project, we significantly improved the BERT model using a few novel NLP techniques.

1. Ensemble BERT with weighting is an effective way of improving the single BERT.
2. Our novel data augmentation algorithm based on synonym replacement is a good way of enriching the diversity of training set. Adding BERT models fine-tuned on augmented data can significantly improve the performance of BERT ensemble.
3. External linguistic knowledge such as grammar rules and common sense is helpful to correcting the predictions of the models.
4. BiDAF can contribute positively to the ensemble's performance despite its significantly lower performance compared to BERT, meaning that it can correct some systematic errors made by BERT models.

In the future, we can probably explore a NMT-based data augmentation, which will introduce more diversity to the data. Also, more linguistic knowledge can be integrated into machine learning models. At last, ensembling different models can overcome the systematic errors of the current system.

References

- Rajpurkar, Pranav, Robin Jia, and Percy Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD." *arXiv preprint arXiv:1806.03822* (2018).
- Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
- Wei, Jason W., and Kai Zou. "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks." arXiv preprint arXiv:1901.11196 (2019).
- Yu, Adams Wei, et al. "Qanet: Combining local convolution with global self-attention for reading comprehension." arXiv preprint arXiv:1804.09541 (2018).
- Mollá, Diego, and Mary Gardiner. "Answerfinder: question answering by combining lexical, syntactic and semantic information." Proceedings of the Australasian Language Technology Workshop 2004. 2004.
- Miller, George A. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.